МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное бюджетное образовательное учреждение высшего образования  
“Тихоокеанский государственный университет”

**ФАКУЛЬТЕТ КОМПЬЮТЕРНЫХ И ФУНДАМЕНТАЛЬНЫХ НАУК**

**Лабораторная работа 5**  
на тему:  
  
Построение 3D-сцены

Работу выполнил  
студент 4 курса

группы ПО(аб)-91  
Высоцкий Кирилл Геннадьевич

Проверила:

Вихтенко Эллина Михайловна

Хабаровск-2022

Цель работы: необходимо реализовать программу, которая отображает трехмерную сцену из множества трехмерных текстурированных объектов.

Вариант 1: Улица из домов разных цветов, размеров и типов.

Исходный код на языке Java Script:

Файл app.js

import Viewer from "./viewer.js";

import \* as THREE from "./three.module.js";

export default class {

    constructor() {

        Viewer.init(

            {

                renderer:{

                    parent:document.body,

                    antialias: true,

                    alpha: true,

                }

            }

        );

        this.createObject();

    }

    createObject() {

        // this.object = new THREE.Mesh(

        //     new THREE.BoxGeometry(1,1,1),

        //     new THREE.MeshStandardMaterial({color:"gray"})

        // )

        // Viewer.scene.add(this.object);

        // this.object.position.z = -1;

        // var that = this;

        // this.object.visible = true;

        this.planeGeometry = new THREE.PlaneGeometry(60,20,1,1);

        this.planeMaterial = new THREE.MeshBasicMaterial({color: "green"});

        this.plane = new THREE.Mesh(this.planeGeometry,this.planeMaterial);

        Viewer.scene.add(this.plane);

        this.plane.rotation.x=-0.5\*Math.PI;

        this.plane.position.x = 15;

        this.plane.position.y = 0;

        this.plane.position.z = 0;

        this.cubeGeometry = new THREE.BoxGeometry(4,4,4);

        this.cubeMaterial = new THREE.MeshBasicMaterial({color: "orange"});

        this.cube = new THREE.Mesh(this.cubeGeometry, this.cubeMaterial);

        Viewer.scene.add(this.cube);

        this.cube.position.x = -4;

        this.cube.position.y = 3;

        this.cube.position.z = 0;

        this.cube2Geometry = new THREE.BoxGeometry(6,6,6);

        this.cube2Material = new THREE.MeshBasicMaterial({color: "blue"});

        this.cube2 = new THREE.Mesh(this.cube2Geometry, this.cube2Material);

        Viewer.scene.add(this.cube2);

        this.cube2.position.x = 8;

        this.cube2.position.y = 3;

        this.cube2.position.z = 0;

        this.cube3Geometry = new THREE.BoxGeometry(8,8,8);

        this.cube3Material = new THREE.MeshBasicMaterial({color: "violet"});

        this.cube3 = new THREE.Mesh(this.cube3Geometry, this.cube3Material);

        Viewer.scene.add(this.cube3);

        this.cube3.position.x = 25;

        this.cube3.position.y = 3;

        this.cube3.position.z = 0;

        this.coneGeometry = new THREE.ConeGeometry(4,4,4);

        this.coneMaterial = new THREE.MeshBasicMaterial({color: "red"});

        this.cone = new THREE.Mesh(this.coneGeometry, this.coneMaterial);

        Viewer.scene.add(this.cone);

        this.cone.position.x = -4;

        this.cone.position.y = 7;

        this.cone.position.z = 0;

        this.cone2Geometry = new THREE.ConeGeometry(6,6,6);

        this.cone2Material = new THREE.MeshBasicMaterial({color: "brown"});

        this.cone2 = new THREE.Mesh(this.cone2Geometry, this.cone2Material);

        Viewer.scene.add(this.cone2);

        this.cone2.position.x = 8;

        this.cone2.position.y = 8;

        this.cone2.position.z = 0;

        this.cone3Geometry = new THREE.ConeGeometry(8,8,8);

        this.cone3Material = new THREE.MeshBasicMaterial({color: "yellow"});

        this.cone3 = new THREE.Mesh(this.cone3Geometry, this.cone3Material);

        Viewer.scene.add(this.cone3);

        this.cone3.position.x = 25;

        this.cone3.position.y = 10;

        this.cone3.position.z = 0;

    }

}

Файл viewer.js

import \* as THREE from "./three.module.js"

export default {

    init(data) {

        this.createRenderer(data.renderer);

        this.createCamera();

        this.createScene();

        this.createLight();

        this.update();

    },

    createRenderer(settings){

        if(this.renderer){

            this.renderer.domElement.parentNode.removeChild(this.renderer.domElement);

            this.renderer.dispose();

        }

        this.renderer = new THREE.WebGLRenderer(settings);

        settings.parent.appendChild(this.renderer.domElement);

        this.renderer.setSize(document.body.offsetWidth, document.body.offsetHeight);

    },

    createCamera(){

        this.camera = new THREE.PerspectiveCamera(

            75,

            document.body.offsetWidth/document.body.offsetHeight,

            0.1,

            1000

        )

        // this.camera.position.set(3,3,3);

        // this.camera.lookAt(0,0,0);

        this.camera.position.x = -5;

        this.camera.position.y = 10;

        this.camera.position.z = 25;

        this.camera.lookAt(0,0,0);

    },

    createScene(){

        this.scene = new THREE.Scene();

    },

    createLight(){

        // this.light1 = new THREE.DirectionalLight(0xffffff,.5);

        // this.scene.add(this.light1);

        // this.light1.position.set(5,5,5);

        this.spotLight = new THREE.SpotLight(0xeeeece);

        this.scene.add(this.spotLight);

        this.spotLight.position.set(1000, 1000, 1000);

        this.spotLight2 = new THREE.SpotLight(0xffffff);

        this.scene.add(this.spotLight2);

        this.spotLight2.position.set( -200, -200, -200);

    },

    update(){

        this.renderer.render(this.scene,this.camera);

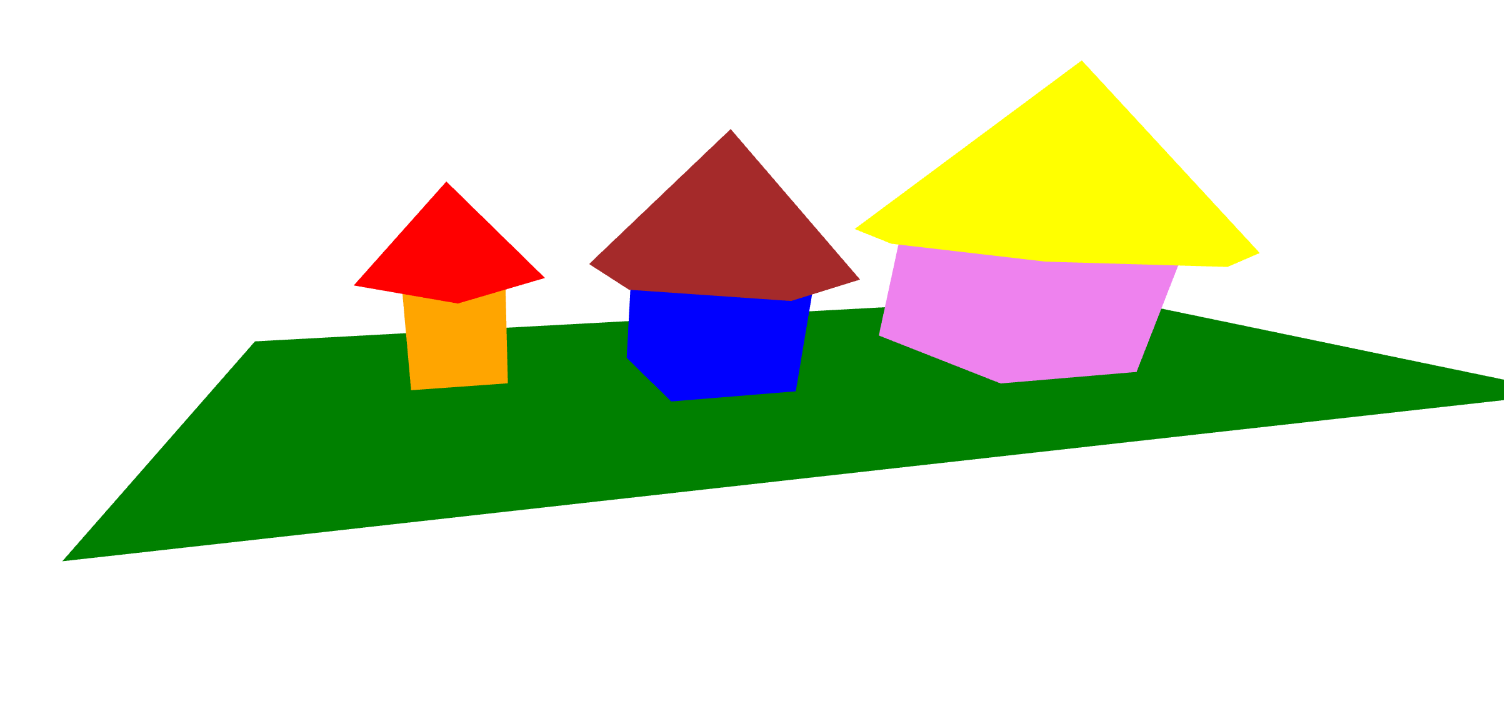
        var that = this;

        requestAnimationFrame( ()=>{that.update();});

    }

}

Результат:



Вывод: в рамках лабораторной работы были изучены основы работы с 3d-графикой и примитивами, а также нарисована 3-d сцена “ Улица из домов разных цветов, размеров и типов ”.